Evolution of Dynamic Analysis
The RSLP Experience

Cedric Pascua & Veronica Vigil
13 September 2012

Agenda

- RSLP Experience
- The Dynamic Analysis Challenge
- Dynamic Analysis Overview
- Software-only and Hybrid Simulations
  - Peacekeeper ICBM – Early ‘80s
  - Small ICBM – Late ‘80s
  - MSLS – Early ‘90s
  - RSLP – Late ‘90s
- Hardware-in-the-Loop Simulations
  - Processor-in-the-Loop – Early ‘00s
  - Computer-in-the-Loop – Mid ‘00s
  - Multiple-Computers-in-the-Loop – Early ‘10s
- Future

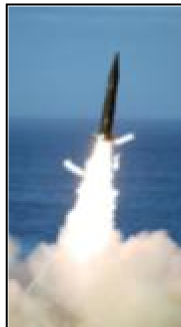
RSLP Experience

- For over 30 years, the USAF has recognized the crucial role of Independent Verification and Validation (IV&V) of command and control (C^2) software
  - InterContinental Ballistic Missiles (ICBM’s)
  - Missile Defense Target vehicles
  - Space launch vehicles [solid and liquid engine]
  - Test/experiment sounding rocket launch vehicles
- The Rocket Systems Launch Program (RSLP) provides a unique IV&V for the USAF incorporating two distinct parts:
  - **Static Analysis** reviews are performed against all C^2 software development products spanning the mission lifecycle
    - Document and code reviews, requirements, design, and test
  - **Dynamic Analysis** executes the software in its native environment with simulated flight environment and dynamics

Launch and Missile Programs History

- 16 consecutive successful launches of new vehicle configurations:
  - STARBIRD
  - LCLV
  - TCMP-I
  - TCMP-II
  - ait-1
  - ait-2
  - NTW/QRLV
  - MSLS
  - TLV
  - Minotaur I/II
  - LRALT
  - MRT
  - NFIRE TLV
  - LV-2
  - Juno
  - Minotaur IV

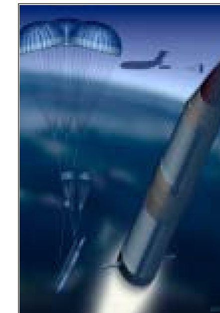
Minotaur IV



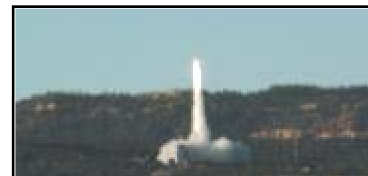
NTW

…With 4 different launch vehicle contractors

- Orbital
- Lockheed Martin
- Coleman Aerospace
- Space Vector



LRALT



ait

… From 8 different Ranges

- Wallops
- Cape Canaveral
- Wake
- Vandenberg
- Kodiak
- PMRF
- RTS
- White Sands

Juno

Minotaur I

TCMP

Experienced and Highly Successful Team With Missile Domain Knowledge

The Dynamic Analysis Challenge

- Execute mission critical code in its native environment and without modification to provide the best prediction of operational performance

- Trade-offs – Fidelity/Visibility, Cost/Schedule
  - Real time vs. non-real time
  - Operational computer hardware vs. simulated computer hardware
  - Operational operating system vs. simulated operating system
  - Vehicle interface hardware vs simulated interface
  - Closed loop vs. open loop

- Constraints
  - Computational capability
  - Availability of operational computers

Dynamic Analysis

Dynamic Analysis Overview

- Software analysis to ensure the correctness and effectiveness of:
  - Software implementation
  - Programmed mission parameters
  - Implementation of interfaces
  - Expected performance in a closed loop simulation
- Relies on an analysis tool set which executes the software under test in its native environment
- Focus analysis priorities based on Lessons Learned and mission risk
  - Lessons from previous analyses and missions reviewed prior to each new analysis
  - Risks are defined from:
    - Reviewing requirements
    - Considering heritage of equipment and software
    - Evaluating changes from similar efforts

Dynamic Analysis Overview

- Evolved from scientific, software-only simulations to more complex Hardware-in-the-Loop (HWIL) simulations using operational computers as technology allows
  - Software-only
    - High fidelity flight dynamics and environment modeling
    - Simulated vehicle/computer interfaces
    - Simulated operating system
    - Non-real time
    - Extensive breakpoint capability
  - HWIL
    - High fidelity flight dynamics and environment modeling
    - Actual vehicle/computer interfaces
    - Actual operating system
    - *Actual operational computer/processor*
    - Real time
    - Limited or no breakpoint capability

Software-Only and Hybrid Simulations

Software-Only Simulation – Peacekeeper ICBM Early ‘80s

- Problem – Availability of operational ICBM computer assets extremely limited
- Solution – Simulate the performance of the ICBM operational computer on a mainframe
  - Software Interpretive Computer Simulation (ICS) simulates the instruction set architecture of the target computer
  - Executes each binary, machine code, instruction by performing the exact register transactions

Photo courtesy of LLNL ©

Software-Only Simulation – Peacekeeper ICBM Early ‘80s

- Advantages
  - High fidelity simulation of code execution
  - Extensive breakpoint and trace capability
- Disadvantages
  - Non-real time
  - Development cost is extremely high
  - Limited applicability
  - Instruction execution times estimated

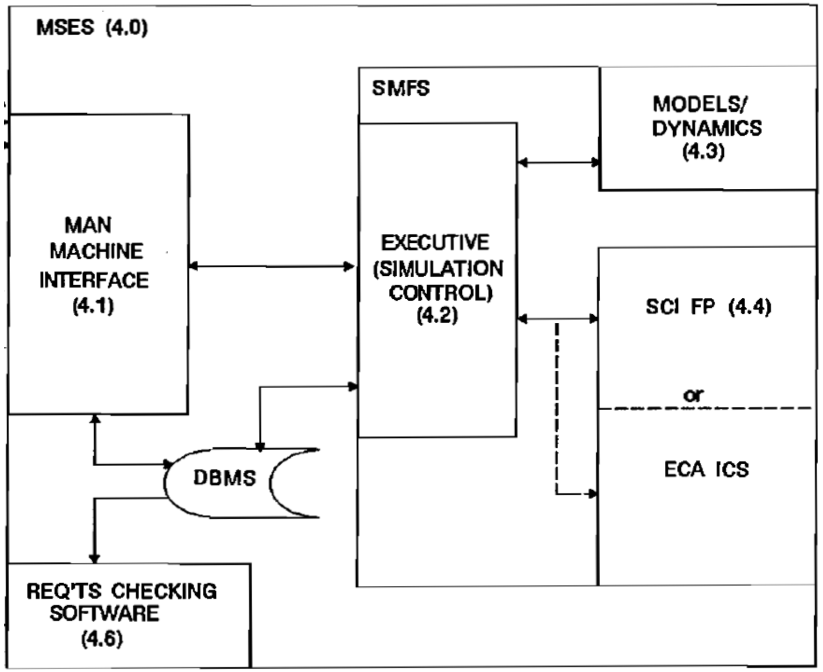
Photo courtesy of LLNL ©

Hybrid Simulation – Small ICBM Late '80s

- Problem – Extremely high ICS development costs
- Solution – Utilize a computer of the same architecture as the target 1750A based computer
  - Hardware ICS provides the actual instruction set architecture of the target computer
  - Executes each binary, machine code, instruction by performing the exact register transactions
  - Specialized 1750A computer in conjunction with unique ICS software supplied breakpoint and diagnostic tools
    - Discrete detection
    - Monitor, trace, and tag memory accesses
    - Save and execute from breakpoint

Hybrid Simulation – Small ICBM Late ‘80s

- Advantages
  - Less software ICS development
  - Same or better fidelity as software ICS implementation
  - Reduced cost of development of instruction set and diagnostics
  - Better simulation of instruction execution times
- Disadvantages
  - Still not real-time
  - Development costs still high
  - Applicability still limited to single application
  - Interface of specialized 1750A computer and dynamic simulation computer proved more difficult than anticipated



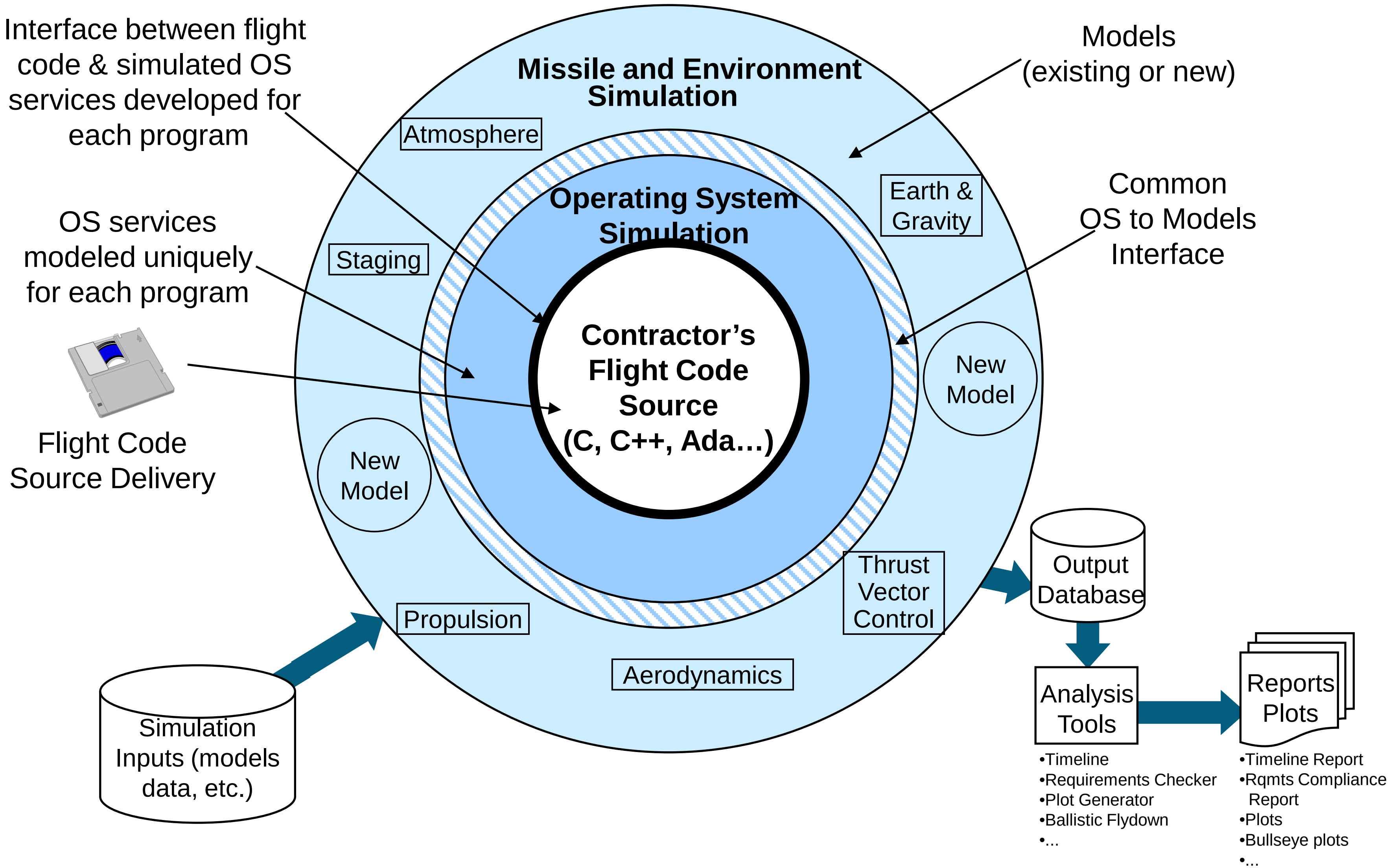
Software-only Simulation (2nd Generation) RSLP – Late '90s

- Problem – Support several concurrent launch systems with multiple computer architectures within tighter schedules and budgets
  - Several computer architectures
  - Multiple launch vehicles with different environments
  - Cost is extremely constrained
- Solution – Reuse a highly modularized software simulation
  - Re-architect to a reconfigurable, adaptable simulation environment
  - Rely on commercially available and existing internal tools
  - Leverage enterprise computing capability
  - Simulate target operating system and computer interfaces

Software-only Simulation (2nd Generation) RSLP – Late '90s

- Advantages
  - Rapid simulation development
  - Applicability to a wide variety of computer architectures
  - Multiple host platforms available
  - Full control of code execution
  - Runtime access to code
  - Cheaper development costs; shorter schedules; smaller teams
- Disadvantages
  - Requires modification of code under test
    - Memory mappings
    - I/O APIs
    - Operating system calls
  - Not real-time execution
    - Watchdog timers altered
    - No timing analysis

RSLP – Flight Software Test Bed Architecture



RSLP – Flight Software Test Bed Architecture

- Modular, reusable simulation
  - Reconfigurable infrastructure
    - Operating system
    - Computer interfaces
  - Flexible simulation infrastructure
    - Control/sequencing/timing
    - Analysis data collection and output
- Re-hosts C^2 software for different computer environment
  - Translates target operating system functionality to host operating system functionality
  - Adjusts memory mapping and device-specific APIs for host computer
  - Replaces or alters code which is specific to target-computer
  - Cross compiles for execution on host computer

RSLP – Flight Software Test Bed Architecture

- Uses high fidelity 6-degree-of-freedom (6DOF) environment and flight dynamics models developed for separate, GN&C analysis tool
  - GN&C analysis results available for comparison
  - Provides additional GN&C verification
  - Models developed and verified in separate effort

Hardware-In-The-Loop (HWIL) Simulations

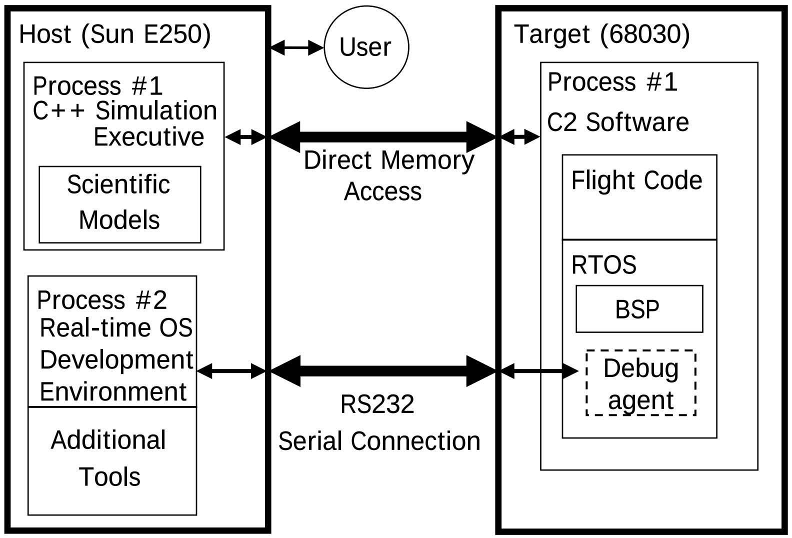
HWIL Simulation Overview

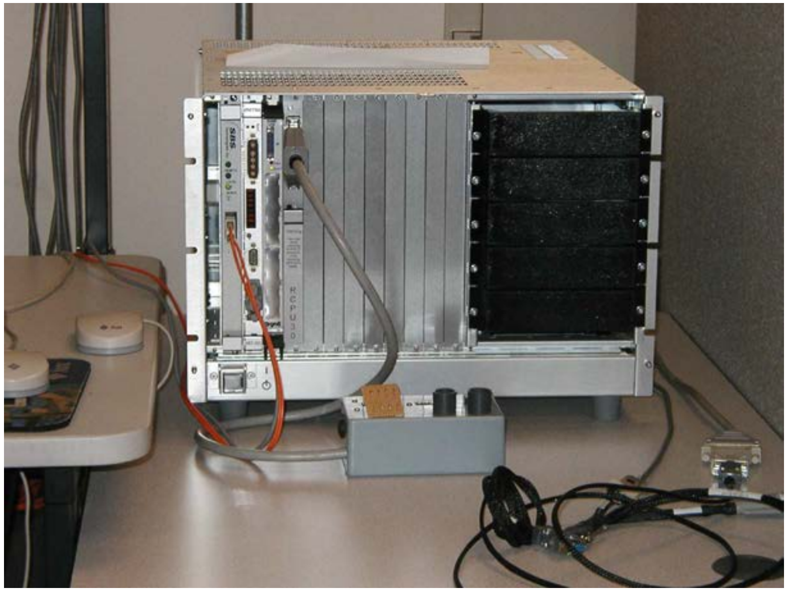
- Utilizes representative operational computer hardware to host C^2 software in native environment
  - Maintains real-time sequencing and processing
  - Responds realistically to operational computer inputs and outputs
- Executes C^2 software in a computer environment which is a replica of target environment
  - Operation or engineering version of operational hardware
  - Identical operating system and firmware
- Relies on mission-specific, high fidelity, closed loop flight dynamics and environment modeling to interact with C^2 software
  - Models interact through real hardware and software interfaces to the operational computer and software

Processor-in-the-Loop
RSLP – Early '00s

- Problem – Execute C^2 software with no modifications and without expensive operational hardware
- Solution – Compromise by simulating operational computer using representative processor
  - Use engineering or commercially available processor
  - Reuse existing highly modularized software simulation with added support for operational processor interfaces
- Advantages
  - Modifies far less code
  - Realistic timing and sequencing
  - Same benefits as modular software simulation
- Disadvantages
  - Modifications to C^2 software to accommodate missing computer interfaces

RSLP – Universal S/W Test Bed Architecture





RSLP – Universal S/W Test Bed Architecture

- Same modular, reusable simulation using high fidelity 6DOF environment and flight dynamics models
  - Updated FSTB architecture
- Utilizes same communication bus as operational processor
  - Simulation communicates with C^2 software using shared memory over bus
- Simulation control and modeling performed by separate host computer
  - Sun workstation connected to flight processor bus via fiber-optic link

Computer-in-the-Loop RSLP – Mid '00s

- Problem – Execute C^2 software in real-time and without modifications
- Solution – Execute C^2 software in operational computer
  - Use flight qualified or engineering version computer
  - Redesign processor-in-the-loop simulation with added support for operational computer interfaces and real-time processing
- Advantages
  - No modifications to C^2 software
  - Realistic timing and sequencing
  - Same benefits as modular processor-in-the-loop simulation

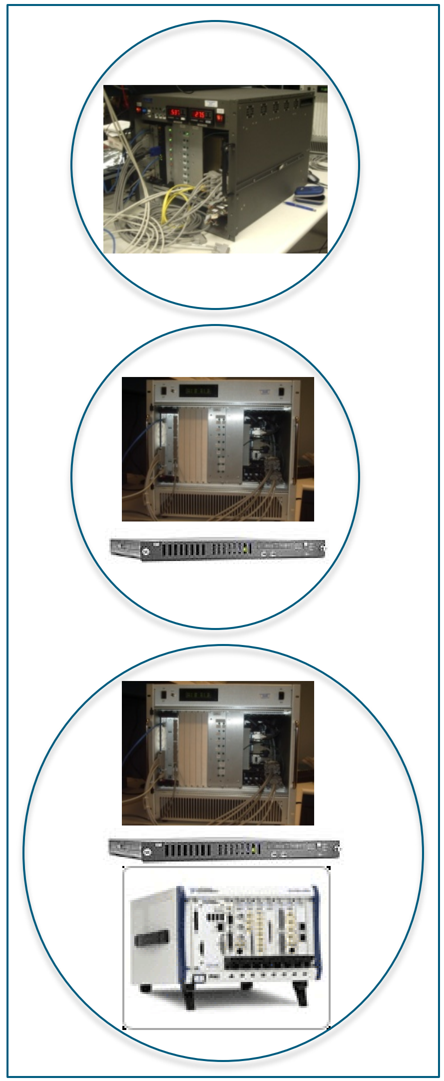
Computer-in-the-Loop RSLP – Mid ‘00s

- Disadvantages
  - Insight to performance through external interfaces only
    - Telemetry
    - Serial, discrete, busses
  - Requires expensive, specialized, high performance computing resources
  - Demands multi discipline support
    - Cable fabrication and installation
    - Device driver development

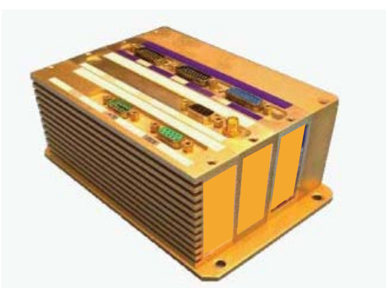
Computer-in-the-Loop Configurations

- Customized configurations of single board computers, I/O boards, I/O systems, workstations, cabling and software for a variety of flight computers

- **MC 680xx**



- **PPC 75x**

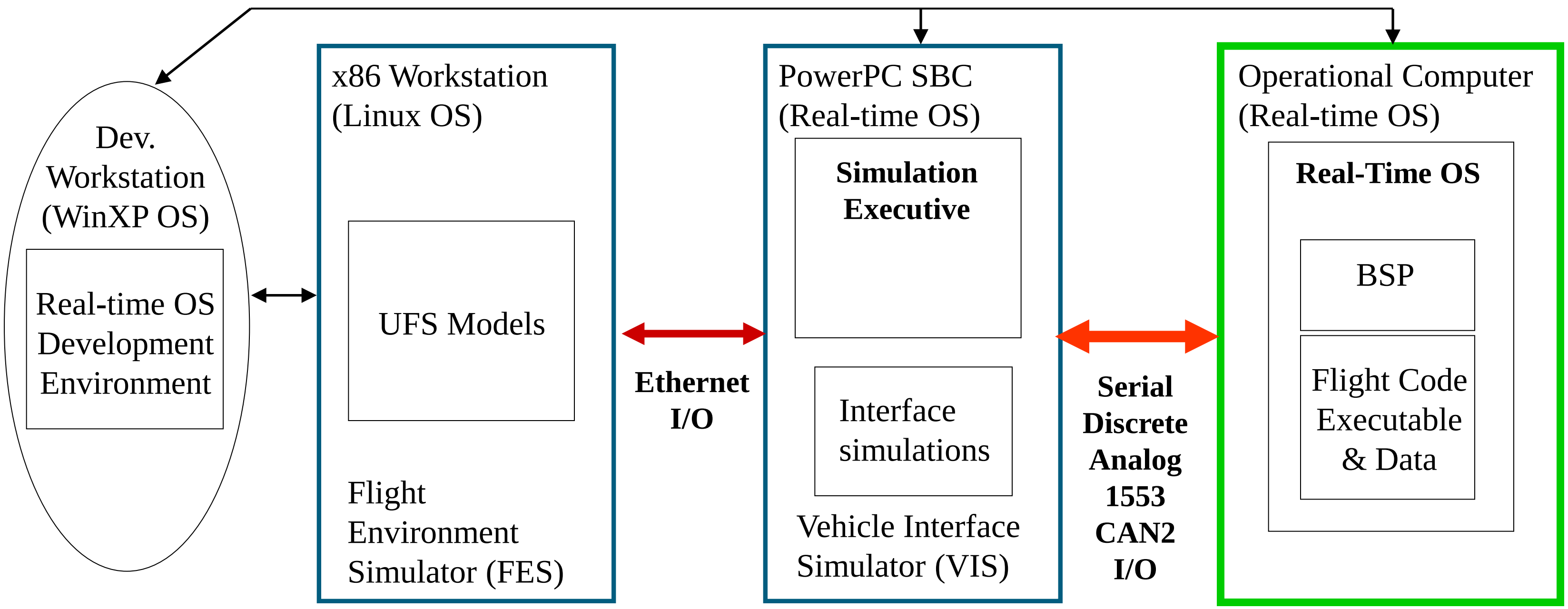


- **PPC 82xx**

- **PPC 74xx**

Computer-in-the-Loop Architecture

- Four software components
  - Development, execution, and analysis station
  - Environment and flight dynamics modeling computers
  - Simulation control and flight computer interface computers
  - Operational computers (with software under test)



Computer-in-the-Loop Equipment

- Flight Environment Simulation Computer
  - Common, inexpensive scientific computers using Linux operating system
  - Same computer hosts GN&C analysis tool (i.e., models)
- Vehicle Interface Simulation Computer
  - Embedded computer boards on VME bus using deterministic, real-time operating systems
  - VME I/O boards, PCI Mezzanine Card (PMC) boards, I/O subsystems
  - Custom cabling to flight computer
- Operational Computer
  - Government furnished equipment from contractor
  - Customized or COTS
  - Flight qualified or engineering-development-unit (EDU) grade

Multiple-Computers-in-the-Loop
RSLP – Early '10s

- Problem – Simulate performance of distributed computing environment in real-time and without modifications
- Solution – Extend existing architecture to accommodate multiple target computers
  - Use operational or engineering flight computers
  - Augment same highly modularized computer-in-the-loop simulation with additional equipment and multi-computer functionality
- Advantages
  - No modifications to flight software
  - Realistic timing and sequencing
  - Same benefits as modular computer-in-the-loop simulation
- Disadvantages
  - Added complexity
  - Additional equipment

Dynamic Analysis Future

- Remain focused on HWIL tool configurations
  – Execute entire suite of operational software in realistic environment
- Improve deterministic, real-time processing
  – Better equipment and software
  – Optimize modeling
- Integrate multiple operational computers
  – Coordinate simulated environments for asynchronous computers
- Add In-Circuit Emulator capability
  – Monitor lowest-level processing
- Improve tool flexibility and ease of use
  – Data driven software re-configuration for each mission
  – Switch driven hardware re-configuration
  – Better reusable, interchangeable hardware and software modules